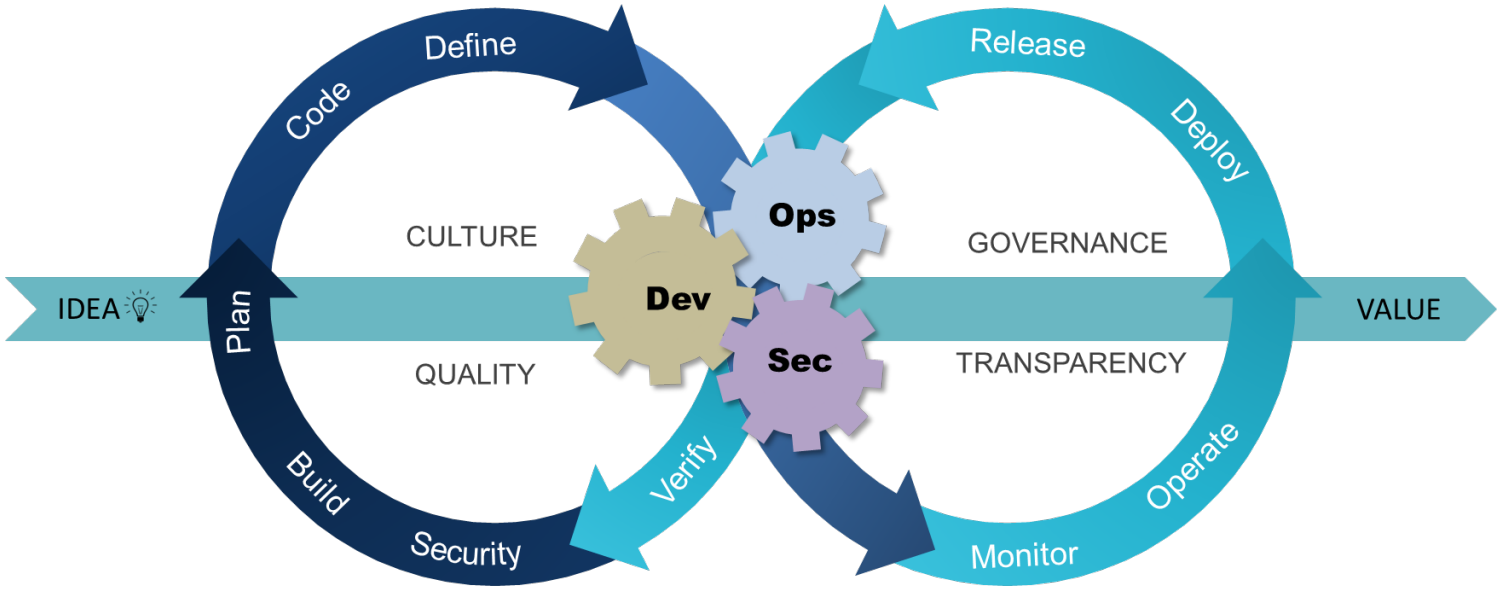


Roadmap to DevSecOps Adoption @ EPA

April 2020

EPA Information Technology Architecture Program

Presentation Agenda

Highlights of the DevSecOps Paper including strategies for near term implementation.

01
Current State

Interviews with EPA Product Teams

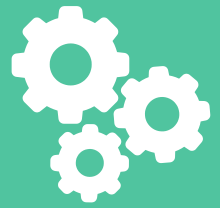
02
Common Takeaways

Requirements, Processes, and Systems' Needs

03
Future State

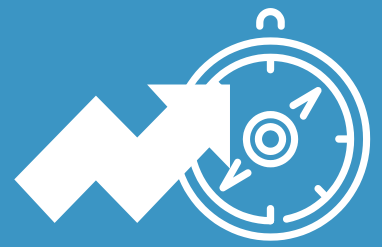
Frictionless DevSecOps Platform using Containers for Product Teams

04
Realize the Teams through Alignment



Create alignment of the new teams via platform, technologies, process and culture

05
Next Steps



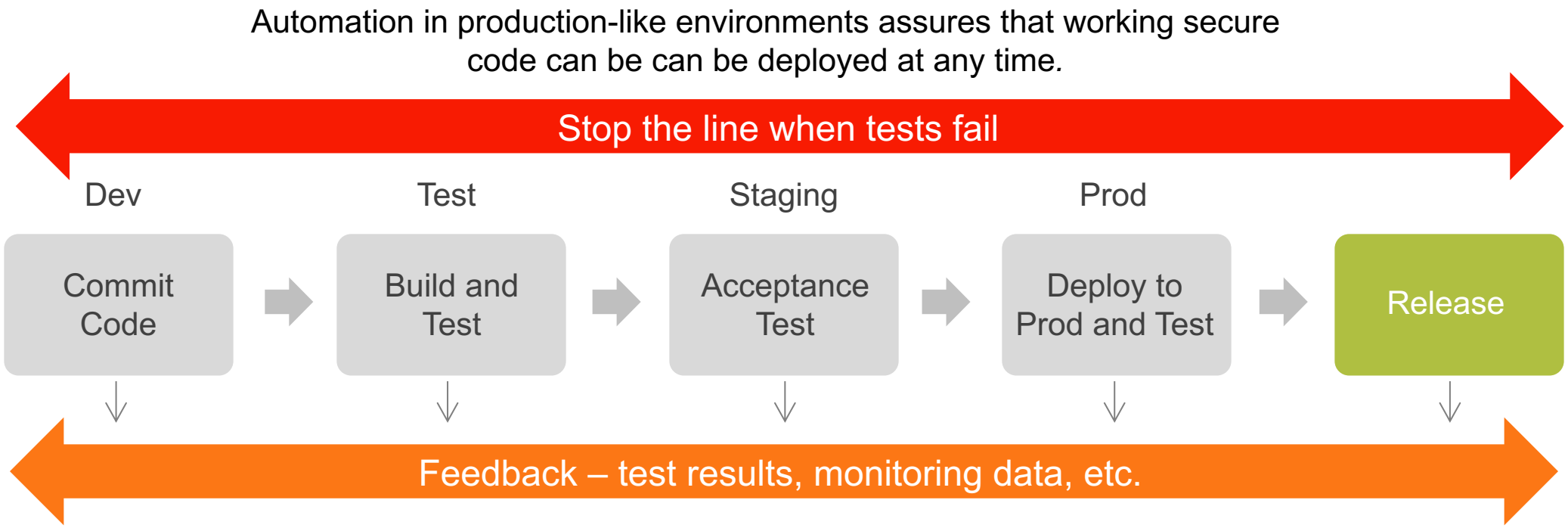
Learn by Doing as Agile teams - Incrementally building Proof of Concepts

What is DevOps?

DevOps (bringing Software Development and Operations together) is a cultural mind set that uses tools and processes to foster two main goals:

1. Continual experimentation in short cycles, taking risks, and learning from failure
2. Understanding that repetition and practice is the prerequisite to mastery

When security is added into the lifecycle it is known as **DevSecOps**.



A ***Deployment*** is the installation of a version of software to an environment (e.g., promoting a new build to production).

01

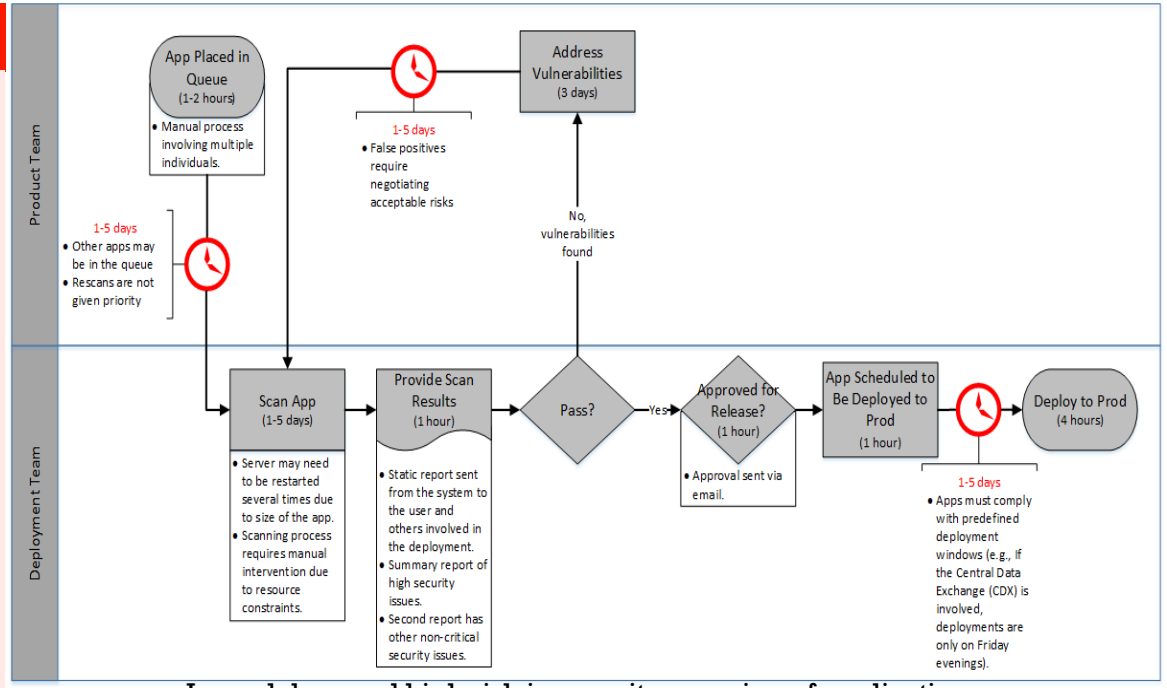
Current State

Engaging The Product Teams

Understanding the current state of delivery and operations at EPA, including the Product Teams' perspectives and their needs to support delivery requirements.

Interviews with Product Teams

- 9 EPA-teams interviewed-comprised of multiple, cross-functional, small agile teams
- Mature in Agile and in Continuous Integration and Continuous Delivery (CI/CD) skills
- All have:
  - Pipelines in AWS or cloud.gov
  - Similar tech stack and open source tools
  - Similar deployment, testing and security processes
  - Similar roles and responsibilities distributed between EPA and contractor team members



Long delays and high risk in security scanning of applications

Operation Concerns

- Address **security scanning** much earlier in the process, which is the nature of DevSecOps
- Reduce the **manual processes** that are involved with the deployment of the artifacts
- Dependency on **specialized skills** to perform the deployment impedes deployments

Physical and Policy Impediments

- No sandbox testing environments for external users that mimics the CDX data flow to NCC
- EPA polices and firewall rules are cost prohibitive to create secondary data flows for users for testing new features
- Staging environments do not match production environments - user experience for testing is not ideal

02
Common Takeaways

Product Teams want DevOps
~~However~~

What the Platform Teams Want

Deliver Software Quicker, Reliably and Securely:
- Cloud platforms and DevOps environment that match the demands of customer requirements
- Deliver on demand
- Separate deployment from release
- Do blue/green releases

Build a Community of Learning:
- Spread learning through, dojos coaching and training
- Communities of Practice and Proofs of Concepts

Cloud is a Differentiator:
- Scale pipelines and create predictable delivery

Replace the ADC Process with an Automated Deployment Process:
- Want a higher predictability in delivery
- Greater transparency
- More frequent releases (Daily/Weekly)
- Acknowledged the necessity of appropriate control gates

Autonomy:
- Create and manage own environments, tools, pipelines
- Schedule their own releases

DevOps does not Scale Easily

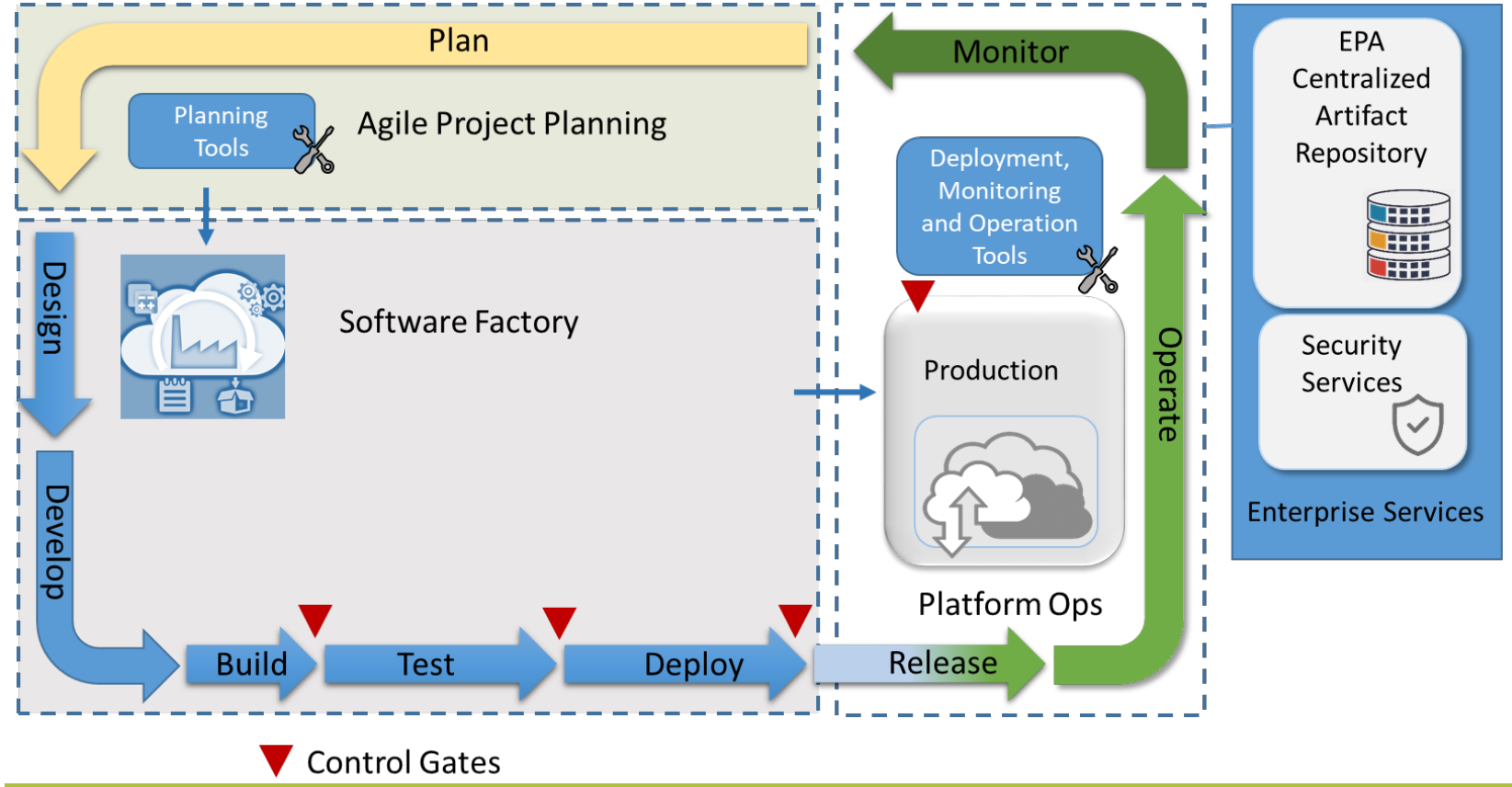
Applications have Dependencies:
- Multiple applications running on the same machine share libraries and components
- Difficult to upgrade software due to conflicts
- Hard to isolate without impacting other services
- VMs do not solve issue as OS being upgraded is shared

CI/CD does not scale well do to complexity:
- Tightly coupled applications use various versions of databases, webservers and OS
- Environments get out of sync; ops team must patch and maintain
- Abundance of SMEs needed to maintain
- Struggle to stay ahead of the security curve for all these new technologies

Security is a late concern:
- Latency with reviewing the apps for security vulnerabilities
- High risk of discovering issues or false positives - must be negotiated, corrected, rebuilt, and rescanned.

03
Future State

Pipelines, Containers and the Software Factory



Key Features

Autonomous Product Teams

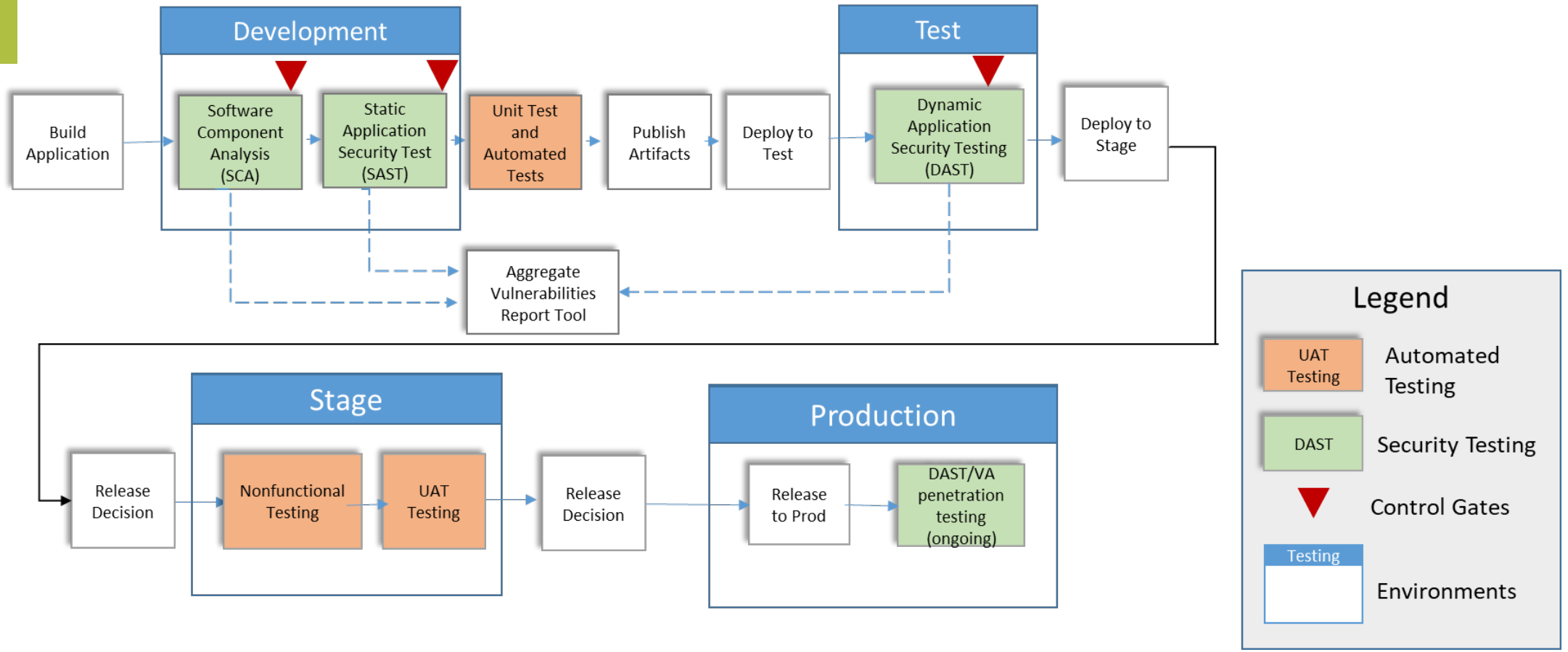
- Plan Releases using Agile Tools
- Deploy everything inside containers (excepts database)
- Ensure consistency across all environments
- Reduce/remove manual deployment
- Shift responsibility of tech stack to Product Team
- Platform Ops team is responsible for everything outside of the container – i.e. the host Ecosystem

Team Needed

- Product Teams use tools, processes, containers and pipelines to manage all aspects of DevSecOps, including security
- Platform Teams provide any and all services to assist the Product Team
- Enabling Teams provide special services and capabilities as needed for the Product Teams
- Complex Sub-System Teams provides deep technical support for creating the environment

03
Future State

Product Teams using DevSecOp Pipelines



03
Future State

Tools, Practice and Services

Activity/Practice	Phase	Why?	Frequency	Tool(s)	Service Provider	Service Consumer
Plan features and capabilities for a release	Plan	Release Management	Every 2 weeks	JIRA, Confluence Slack	Enabling Team	Product Team
SCA – Software Component Analysis	Develop	Identify Dependencies	As needed for a release	Sonatype – Repository Manager, Nexus IQ	Platform Team	Product Team
Container Selection	Develop	Appropriate container base image for application	At start of new feature or product	Docker EPA Centralized Artifact Registry (EPA CAR) (hosted in GitLab)	Enterprise Services	Product Team
SAST – Complete code base check of integrated code	Build	Code Vulnerabilities	Daily/ weekly	Contrast Security	Product Team	Product Team
Container Image Scan	Test	Conduct container image scan OS-check	On Deploy	Twistlock	Platform Team	Platform Team
Aggregate Vulnerabilities Report	Release	Provide findings from various scanning tools	On Deploy	Code DX	Platform Team	Product Team

03
Future State

Tools, Practice and Services

Activity/Practice	Phase	Why?	Frequency	Tool(s)	Service Provider	Service Consumer
Complete System DAST	Deploy	Onboarding to Host Ecosystem	Once per system release	Sonatype – Nexus IQ	Platform Ops	Platform Teams
Track and visualize metrics	Operate	Understand the metrics of the data	Ongoing	Grafana	Production Ops	Product Teams
Production Scanning and Assessment (DAST/VA)	Monitor	Application Security Assessment	Continuous or as required by security team	Contrast Security - Protect	Cloud Application Security Team	Production Ops
Monitor Events	Monitor	Event alerts that integrates with Grafana	Ongoing	Prometheus	Production Ops	Product Teams
CI/CD Pipeline Management	Software Factory	Automate the deployment of containers with integrated security	Daily	GitLab	Platform Team	Product Team
Container Orchestration	Release	Manage the release, management, scaling, networking, and availability of container-based applications.	Daily	AWS EKS or Rancher	Platform Team	Platform Team

04
Realize the Teams through Alignment

Frictionless Support Teams

Product Teams (Developers, Testers)

- Aligned to a single, valuable stream of work, such as a product, service or a set of features
- Empowered to design, build and deliver customer value as quickly and securely as possible without having to handoff work to other teams
- Have complete autonomy over the deployment of their products in any environment (dev, test, stage, prod)

Production Ops Team (Infrastructure, Hosting, Services)

- Supports release, operate, and monitor phases of the DevSecOps lifecycle
- Goal is to make the autonomous Product Teams make use of the platforms to deliver features at a higher pace with reduced coordination and little friction
- Ensure the host ecosystem is secure through Container Orchestration Management and Deployment operations

Enabling Team (Cloud Service Reps, Assisted Services)

- Onboarding and setup new Cloud accounts for Product Teams
- Provide special services and capabilities as needed for the Product Teams
- Team of cloud representatives that provide assisted services that help Product Teams get code through the pipeline process.

Complex subsystem Team (New services that need to be figured out)

- Deep technical support for creating and maintaining the hosted ecosystem.
- AI, Machine Learning, Analytics

05
Next Steps

Form Agile Teams to Work on Proof of Concepts (POC) and Deploy Continuously

Iterate the Platform, Techniques, Process and Culture

Identify Platform Team

- Identify startup Platform Ops Team
- Select Product Team for Proof of Concept (POC).
- Define performance metrics for outcomes

Identify Technologies

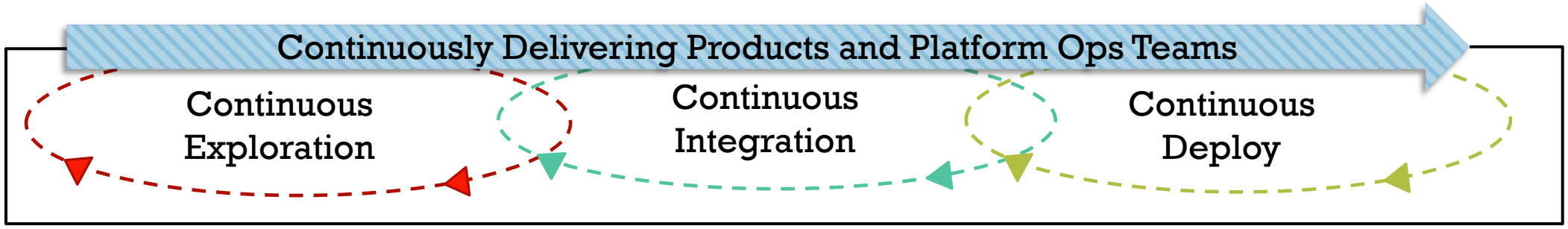
- Create Thinnest Viable Platform (TVP) to support the POC
- Prioritize the creation of Continuous Application Delivery Pipeline
- Deploy tech to run as an Agile DevOps process

Identify Platform Services and Process

- Deploy new services continuously
- Separate ops concerns in the platform by roles and tasks
- Iterate on services needed
- Define SLAs

Define Platform Culture

- Share learning and spread practice
- Actively practice retrospectives
- Experiment with ideas, share in failure, celebrate team success
- Market the platform



Questions?